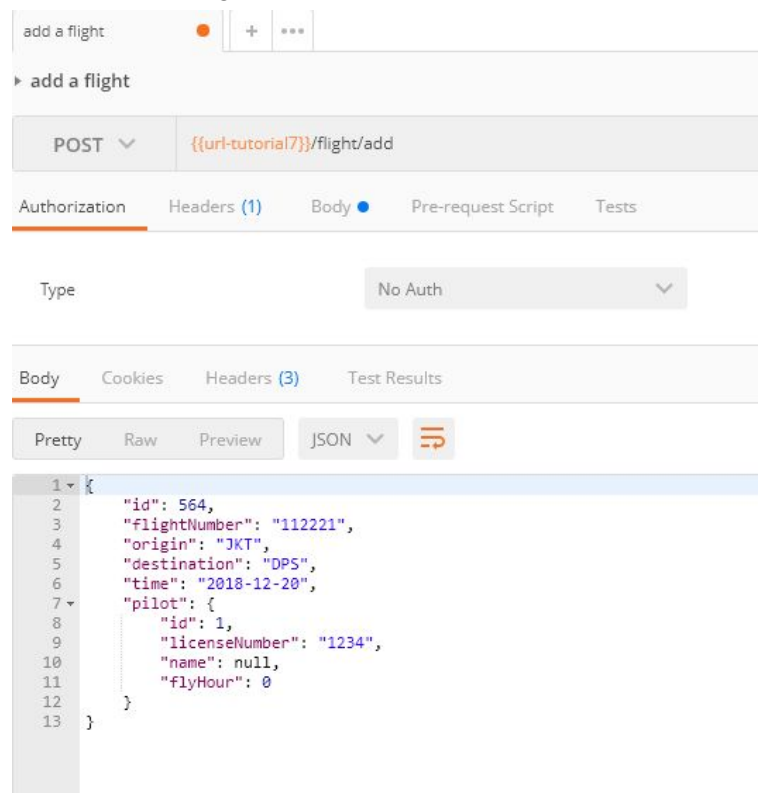


Pada tutorial kali ini, saya mempelajari bagaimana cara membuat web service dari database sendiri, serta membuat dan menggunakan web service yang disediakan oleh penyedia API tertentu. Semuanya dipelajari dengan menggunakan postman, dan dependency tambahan berupa jackson-databind.

1. Pada latihan nomor 1 saya perlu mengubah flight controller yang sebelumnya dibuat menjadi rest controller. Untuk request dan response nya mengikuti yang disoal. Kemudian membuat request di postman sesuai yang diminta soal, umumnya hanya bentuk path yang diinginkan/ditulis di code harus sesuai dengan request yang dibuat, termasuk parameter-parameternya.
 - a. Membuat add flight test di postman



b. Update flight

view all update flight + ***

► update flight

PUT `{{url-tutorial7}}/flight/update/2?destination=DPS&origin=JKT&time=1998-04-23`

Authorization Headers Body Pre-request Script Tests

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value
New key	Value

Body Cookies Headers (3) Test Results

Pretty Raw Preview Text

```
1 flight update succes
```

c. View a flight

► view a flight

GET `{{url-tutorial7}}/flight/view/1151`

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 {
2   "id": 2,
3   "flightNumber": "1151",
4   "origin": "JKT",
5   "destination": "DPS",
6   "time": "1998-04-23",
7   "pilot": {
8     "id": 1,
9     "licenseNumber": "1234",
10    "name": "Coki",
11    "flyHour": 139
12  }
13 }
```

d. View all

view all

GET `{{url-tutorial7}}/flight/all`

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": 2,
4     "flightNumber": "1151",
5     "origin": "JKT",
6     "destination": "DPS",
7     "time": "1998-04-23",
8     "pilot": {
9       "id": 1,
10      "licenseNumber": "1234",
11      "name": "Coki",
12      "flyHour": 139
13    }
14  },
15  {
16    "id": 3,
17    "flightNumber": "1176",
18    "origin": "TIM - Bandar Udara Internasional Mozes Kilangin, M",
```

e. Delete a flight

delete a flight

Add a description

DELETE `{{url-tutorial7}}/flight/delete/2`

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (3) Test Results

Pretty Raw Preview Text

```
1 Flight has been deleted
```

2. Pada latihan kedua, saya perlu mendapatkan api key dari list airport dengan login terlebih dahulu, kemudian membuat rest controller yang nantinya akan mengembalikan list airport yang dikirimkan oleh API tersebut. Setelah itu mengetesnya di postman. Dalam membuat restcontroller kita perlu menggunakan resttemplate yang disediakan oleh spring untuk dapat mengakses rest class http api nya.

```
@RestController()
@RequestMapping("/producer")
public class ProducerController {
    @Autowired
    RestTemplate restTemplate2;

    @GetMapping(value="/listAirport")
    public String poststatus(@RequestParam("city") String city) throws Exception{
        String path = Setting.producerUrl + "&term=" + city;

        return restTemplate2.getForEntity(path, String.class).getBody();
    }
}
```

producer list airport

GET

{{url-tutorial7}}/producer/listAirport?city=surabaya

Params

Send

Authorization

Headers

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Text



Copy Search Save

```
1 [
2   {
3     "value": "SUB",
4     "label": "Surabaya - Juanda International Airport [SUB]"
5   }
6 ]
```